Predicting ratings in multi-criteria recommender systems via a collective factor model

Ge Fan
gefan@tencent.com
Lightspeed & Quantum Studios, Tencent Inc.
Shenzhen, Guangdong, China

Chaoyun Zhang
vyokkyzhang@tencent.com
Lightspeed & Quantum Studios, Tencent Inc.
Shenzhen, Guangdong, China

Junyang Chen*
junyangchen@szu.edu.cn
College of Computer Science and Software Engineering, Shenzhen University
Shenzhen, Guangdong, China

Kaishun Wu
wu@szu.edu.cn
College of Computer Science and Software Engineering, Shenzhen University
Shenzhen, Guangdong, China

ABSTRACT

In a multi-criteria recommender system, users are allowed to give an overall rating to an item and provide a score on each of its attribute. Finding an effective method to exploit a user's multi-criteria ratings to predict the overall rating becomes one of the most important challenges. Among traditional solutions, most of the architectures are not designed in an *end-to-end* manner. These approaches initially estimate a user's multi-criteria scores, and train a separate model to predict the user's overall rating. This introduces extra training overhead, and the overall prediction accuracy is usually sensitive to its multi-criteria ratings models.

In this paper, we propose a collective model to predict user's overall rating by automatically weighting each of the predicted multi-criteria sub-scores. The proposed architecture integrates the multi-criteria ratings and the overall rating models in a unified system, which allows to train and perform multi-criteria recommendation in an end-to-end manner. Experiments on 3 real datasets show that our proposed architectures achieve up to 13.14% lower prediction error over baseline approaches.

CCS CONCEPTS

• **Information systems** → *Information retrieval*; **Recommender systems**; *Collaborative filtering*.

KEYWORDS

Latent factor model, Multi-criteria recommender system, Collaborative filtering, Collective matrix factorization.

ACM Reference Format:
Ge Fan, Chaoyun Zhang, Junyang Chen, and Kaishun Wu. 2021. Predicting ratings in multi-criteria recommender systems via a collective factor model. In *DeMaL@WWW '21*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/1122445.1122456

*Corresponding author.

WWW '21 Companion, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Recent research on the multi-criteria recommender system allows to estimate the overall score of an item by weighting the sub-score of each of its attribute, to reflect the user's preference of the item. As it weights an item using multi-dimensional perspectives, the multi-criteria recommender systems perform superiorly over single-criterion counterparts in terms of the predictive accuracy, and are used more frequently in both industry and academia (e.g.*BeerAdvocate* [1, 20, 32] *TripAdvisor Yahoo!Movies* [2, 17])

One of the most important challenges of the multi-criteria recommender system is integrating the user's multi-criteria ratings to predict the overall rating of the item. To this end, traditional methods calculate the similarity of users with respect to each criterion of an item, and subsequently average the scores to reflect the overall similarity between users [8]. The overall rating of an item is then obtained from a user-based collaborative filtering framework. However, the similarity-based method usually suffers from the poor inference efficiency, as the similarities need to be computed over all user pairs. The complexity grows further with the multi-criteria recommender system, as multiple attributes of the item need to be considered.

Traditional multi-criteria recommender systems usually predict overall rating of an item to a user in two separate stages (e.g. [1], [18]). They first predict ratings of each attribute of the item, and employ a separate model to integrate those scores to obtain an overall rating. However, the overall rating heavily relies on the predicted sub-scored of each attribute, which is particular difficult to predict. As a result, inaccurate prediction of the sub-scores will significantly affect the overall system, lowering the performance of the entire recommender systems.

To overcome this issue, we propose an end-to-end multi-criteria rating system, to integrate the sub-scores and overall rating into a unique architecture. This reduces the dependency of its sub-process. Specifically, we employ a latent factor model [22] to jointly learn users' overall score as well as their preferences on each criterion of an item. Sub-scores of each criterion are treated as latent variables of the model, which are hidden and do not need to be estimated in a separate process. This makes the overall system most robust and accurate, as it does not heavily rely on the predicted score of a specific criterion. Overall, this paper makes three important contributions, namely:

(1) We propose an end-to-end collective factor model to unify the learning processes of sub-scores and overall rating. This reduces the dependency of the score of each attribute, improving the robustness of the entire system;
(2) We investigate the impact of different latent vector sharing schemes between users and items. Experiments show that keeping both the user latent vector and the item latent vector independent outperforms over other two sharing methods;
(3) We compare our proposed architectures with several baseline methods on three real-world datasets. The results show that our solution outperforms baseline approaches by achieving up to 13.14% lower prediction error over state-of-the-art approaches.

2 RELATED WORKS

2.1 Multi-criteria recommendation approaches

A primary method to exploit the information of multi-criteria ratings is extending the user similarity calculation from single-criterion to multi-criteria. Nilashi et al. [19] exploited a fuzzy method for the calculation of similarity between users by employing the multi-criteria ratings. Kermany et al. [10] integrated the fuzzy cosine and Jaccard similarity to obtain the final similarity between users/movies.

Recently, two-stage based methods emerges and become more popular in the multi-criteria recommender system. In the first stage, the multi-criteria ratings of the target item are estimated. The overall ratings are obtained by learning the weights of each sub-score by a separate model, such as linear regression [1], support vector regression [8, 32] and neural networks [11, 18].

2.2 Collective matrix factorization based approaches

The collective matrix factorization (CMF) can be employed to incorporate the rating data and the auxiliary data [21, 24], when the auxiliary data is available. The collective factorization method simultaneously co-factorizes a variety of matrices when an entity participates in multiple relations. It has demonstrated superiority when integrating diverse auxiliary resources, such as social networks [3, 15], geographical information [29–31] and contents of items [23, 24], as it can effectively embed those rich resources [4, 13, 23].

Singh et al. [24] proposed a collective matrix factorization model to improve the predictive accuracy by integrating multiple matrices. Based on the collective matrix factorization, Liu et al. [14] incorporated both explicit and implicit feedback of users to improve recommendation quality. Similarly, Yuan et al. [3, 27] exploited collective matrix factorization method to jointly model data from different sources, which improves the performance of the model.

2.3 Limitations of traditional methods

Although the user similarity calculated by multi-criteria ratings is in general more accurate, such methods usually suffer from the poor efficiency and low robustness, as most of existing multi-criteria based models are not trained in an end-to-end manner. Since it heavily relies on the the sub-scores on different attributes, the overall accuracy of the model is highly sensitive to the its sub-processes.

To resolve this problem, we propose a collective factor model, which combines contributions of overall ratings and multi-criteria ratings in a linear manner. The experimental results demonstrate that our method is superior over existing multi-criteria based approaches in terms of accuracy of overall rating predictions.

3 PRELIMINARIES

Without loss of generality, individual overall ratings can be represented as a weighted adjacent matrix $\mathbf{R}^{(0)} = \{r_{ui}^{(0)}\}_{n\times m}$, where $r_{ui}^{(0)}$ is the overall score that user u rates the item i. n and m are the number of users and items in the system, respectively. Similarly, an adjacent matrix $\mathbf{R}^{(\alpha)} = \{r_{ui}^{(\alpha)}\}$ is be used to represent users' ratings on criterion α. The task of a multi-criteria recommender system is estimating a user's overall rating by utilizing the ratings on each criterion.

3.1 Latent factor model

In this paper, the latent factor model is adopted to learn users' preferences, jointly mapping users and items to the same space with dimensionality k. We denote two latent vectors x_u and y_i, as the preference of user u and the attribute of item i respectively. The predictive score can be easily obtained by the inner product of these two latent vectors:

$$\hat{r}_{ui} = \mathbf{x}_u^T \mathbf{y}_i. \tag{1}$$

The equation (1) can be extended as:

$$\hat{r}_{ui} = \mu + b_i + b_u + \mathbf{x}_u^T \mathbf{y}_i, \tag{2}$$

where b_u and b_i are the biases of user u and item i, respectively, and μ is the global average rating.

We choose the biased matrix factorization (BMF) as our base approach for the following reasons. First, it outperforms traditional recommendation algorithms such as collaborative filtering, basic matrix factorization and probabilistic matrix factorization in terms of rating prediction [2, 6]. Second, it has comparable complexity with the basic matrix factorization [6].

4 MODEL ARCHITECTURE

In this section, we introduce the overall framework of the proposed collective factor model (CFM) employed for the multi-criteria recommender system. Our method simultaneously co-factorizes the overall rating matrix and multi-criteria rating matrices, which contributes the overall performance.

4.1 Latent factor model for overall ratings

With the overall rating matrix $\mathbf{R}^{(0)}$, the biased matrix factorization model can be learned by minimizing the loss function:

$$\min_{\mathbf{x}^*, \mathbf{y}^*, b^*} \sum_{u,i} (r_{ui}^{(0)} - \hat{r}_{ui}^{(0)})^2 + \lambda_0 (\|\mathbf{x}_u^{(0)}\|^2 + \|\mathbf{y}_i^{(0)}\|^2 + (b_u^{(0)})^2 + (b_i^{(0)})^2), \tag{3}$$

where λ_0 is the regularization parameter. $\mathbf{x}_u^{(0)}$, $\mathbf{y}_i^{(0)}$, $b_u^{(0)}$ and $b_i^{(0)}$ are model parameters with respect to overall ratings. $\mathbf{x}^*$, $\mathbf{y}^*$ and b^* are general forms of $\mathbf{x}_u^{(0)}$, $\mathbf{y}_i^{(0)}$, $b_u^{(0)}$ and $b_i^{(0)}$. $\hat{r}_{ui}^{(0)}$ is the predictive score defined as follow:

$$\hat{r}_{ui}^{(0)} = \mu^{(0)} + b_i^{(0)} + b_u^{(0)} + (\mathbf{x}_u^{(0)})^T \mathbf{y}_i^{(0)}. \tag{4}$$

We denote the parameter set of the latent factor model as $\mathrm{S}^{(0)}$ for overall ratings.

4.2 Latent factor model for multi-criteria ratings

By using the rating on the criterion α, we can also train a biased matrix factorization model by minimizing the loss function:

$$\min_{\mathbf{x}^*,\mathbf{y}^*,b^*} \sum_{u,i} (r_{ui}^{(\alpha)} - \hat{r}_{ui}^{(\alpha)})^2 + \lambda_\alpha (\|\mathbf{x}_u^{(\alpha)}\|^2 + \|\mathbf{y}_i^{(\alpha)}\|^2 + (b_u^{(\alpha)})^2 + (b_i^{(\alpha)})^2), \tag{5}$$

where $\mathbf{x}_u^{(\alpha)}$, $\mathbf{y}_i^{(\alpha)}$, $b_u^{(\alpha)}$ and $b_i^{(\alpha)}$ are model parameters with respect to ratings on criterion α. $\hat{r}_{ui}^{(\alpha)}$ is the predictive score on criterion α defined as follow:

$$\hat{r}_{ui}^{(\alpha)} = \mu^{(\alpha)} + b_i^{(\alpha)} + b_u^{(\alpha)} + (\mathbf{x}_u^{(\alpha)})^T \mathbf{y}_i^{(\alpha)}. \tag{6}$$

Similarly, we use $\mathrm{S}^{(\alpha)}$ to represent the parameter set of the latent factor model with respect to the rating on criterion α. When multi-criteria ratings are unavailable, only equation (4) is exploited to predict missing values in the overall ratings. In this paper, both overall ratings and multi-criteria ratings are taken into consideration to train the predictive model.

4.3 Collective factor model

When both overall ratings and multi-criteria ratings are available, we combine equation (3) and equation (5) in a unified framework. We assume that the final predictive score of the overall rating $\hat{r}_{ui}$ is a linear combination of the sub-rating of each criterion, with the predicted overall rating, i.e.:

$$\hat{r}_{ui} = \hat{r}_{ui}^{(0)} + \sum_{\alpha=1}^{c} w_\alpha \hat{r}_{ui}^{(\alpha)} + \varepsilon, \tag{7}$$

where w_α is the weight of the predictive score $\hat{r}_{ui}^{(\alpha)}$ on criterion α, and ε is the error variable. The final objective function of the collective factor model includes three parts: the loss of overall ratings, the loss of multi-criteria ratings and the regularization term, defined as:

$$\begin{aligned} L &= L_0 + \sum_{c=1}^{\alpha} \Theta_\alpha L_\alpha + \Phi \\ &= \min_{\mathbf{x}^*,\mathbf{y}^*,b^*} \sum_{u,i} (r_{ui}^{(0)} - \hat{r}_{ui})^2 + \sum_{\alpha=1}^{c} \Theta_\alpha \sum_{u,i} (r_{ui}^{(\alpha)} - \hat{r}_{ui}^{(\alpha)})^2 + \Phi, \end{aligned} \tag{8}$$

where c is the number of criteria and Θ_α is a hyper-parameter which controls contributions of ratings on criterion α. Φ is the regularization term to reduce overfitting:

$$\begin{aligned} \Phi &= \lambda_0 (\|\mathbf{x}_u^{(0)}\|^2 + \|\mathbf{y}_i^{(0)}\|^2 + (b_u^{(0)})^2 + (b_i^{(0)})^2) \\ &+ \sum_{\alpha=1}^{c} \lambda_\alpha (\|\mathbf{x}_u^{(\alpha)}\|^2 + \|\mathbf{y}_i^{(\alpha)}\|^2 + (b_u^{(\alpha)})^2 + (b_i^{(\alpha)})^2 + w_\alpha^2). \end{aligned} \tag{9}$$

We apply the stochastic gradient descent approach train the overall model, to obtain the optimal parameters in equation (8).

A simple way to share the knowledge is utilizing a common latent vector between different users or/and items. We employ three variants of knowledge sharing schemes in our CFM, namely (1) sharing knowledge between overall and all criterion rating for a user (CFM_{user}); (2) sharing knowledge between overall and all criterion rating for a item (CFM_{item}); and (3) sharing knowledge implicitly for a user or an item (CFM_{ind}). We show the structures of three different methods in figure 1. Specifically,

(1) The CFM_{user} shares the user knowledge by forcing $\mathbf{x}_u^{(0)} = \mathbf{x}_u^{(1)} = \ldots = \mathbf{x}_u^{(c)}$, while freeing other parameters independent.
(2) The CFM_{item} shares the item knowledge by keeping $\mathbf{y}_i^{(0)} = \mathbf{y}_i^{(1)} = \ldots = \mathbf{y}_i^{(c)}$, while other user latent vectors are not constrained.
(3) The CFM_{ind} does not apply any constrains on the user latent vector and the item latent vector, while keeping $\mathbf{x}_u^{(0)} \neq \mathbf{x}_u^{(1)} \neq \ldots \neq \mathbf{x}_u^{(c)}$ and $\mathbf{y}_i^{(0)} \neq \mathbf{y}_i^{(1)} \neq \ldots \neq \mathbf{y}_i^{(c)}$. This introduces stronger flexibility to the model.

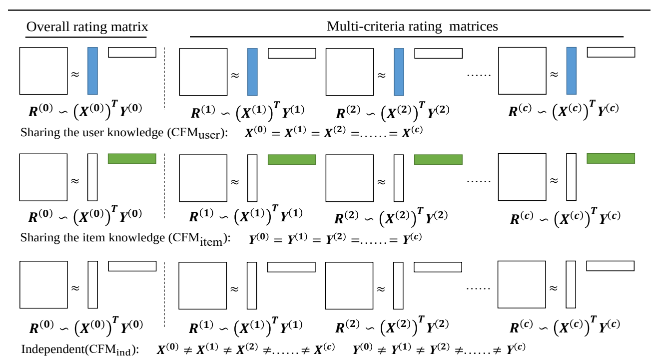

Figure 1: The illustration of CFM method.

5 EXPERIMENTS

In order to evaluate the performance of the proposed architectures, we compare our solution with 8 baseline approaches on 3 benchmark multi-criteria datasets.

5.1 Dataset

The datasets we use include *TripAdvisor*, *Yahoo!Movies* and *BeerAdvocate*. The *TripAdvisor* dataset was released by Wang et al. [26], which consists of 1,725 users, 3,347 items and 29,962 ratings. This dataset in very comprehensive in terms of criteria, i.e. *Service*, *Rooms*, *Sleep Quality*, *Location*, *Cleanliness* and *Value*. The

Yahoo!Movies dataset was released by Jannach et al. [9] which embraces four criteria, including (*Acting*, *Direction*, *Story* and *Visuals*) for users. The *BeerAdvocate* website allows the user to rate four attributes (*Aroma*, *Apprearance*, *Palate* and *Taste*) of beer. By removing those inactive users, the pre-processed dataset includes 3, 238 users, 2, 893 items and 88, 242 ratings. For all datasets, we only reserve those users who have rated more than 10 items, inactive users are less important in the system. We show details of the datasets employed are given in the table 1.

Table 1: The statistics of datasets.

Datasets	#Users	#Items	#Ratings	Sparsity
TripAdvisor	1595	539	10273	98.81%
Yahoo!Movies	1797	1279	39489	98.28%
BeerAdvocate	3238	2893	88242	99.06%

We employ the cross-validation method to evaluate the accuracy of each approach based on five independent instances over training and test set [16, 28]. The training set consists of 80% of the original data and the remaining data is for testing. We select *Mean Absolute Error* (MAE) and *Root Mean Square Error* (RMSE) to evaluate the performance of all models considered:

$$\begin{aligned} MAE &= \frac{1}{|E^P|}\sum_{(u,i)\in E^P}|r_{ui}^{(0)} - \hat{r}_{ui}| \\ RMSE &= \sqrt{\frac{1}{|E^P|}\sum_{(u,i)\in E^P}(r_{ui}^{(0)} - \hat{r}_{ui})^2}. \end{aligned} \tag{10}$$

5.1.1 Baseline methods. We compare our method with 8 baseline methods. Specifically,

- **UserKNN** [1] is the standard user-based collaborative filtering method that has been widely used for the recommander system.
- **MultiUserKNN** [8] calculates the similarity of two users on each criterion, while similarities on all criteria are averaged as the final similarity between these two users. We employ the Pearson correlation to measure similarities of user pairs and top-100 users are selected as the target user's neighbors.
- **Biased matrix factorization (BMF)** [22]. BMF is the baseline model of our method. In this method, we only use the overall ratings to train the model parameters.
- **Multilinear singular value decomposition (MSVD)** [12] is used to integrate explicit and implicit relations among user, item and criterion. The approximation tensor is usually obtained by reserving the largest k-model singular values.
- **Multiple linear regressions (MLR)** [5] applies the multiple linear regression model to study the relationship between the multi-criteria ratings and the corresponding overall rating.
- **Support vector regression (SVR)** [8] trained two support regression models from user- and item- side respectively. It then combines these two models to predict the overall ratings.
- **Criteria-independent contextual model (CIC)** [32] In CIC, the multi-criteria ratings are initially estimated by the context-aware recommendation algorithm. The support vector regression is subsequently applied to predict the overall ratings.
- **Deep multi-criteria collaborative filtering (DMCF)** [18] is a two-stages based approach. It first employs a neural network to estimate a user's multi-criteria ratings, then uses a different neural network to predict the overall rating.

5.2 Results and analysis

We show the performance of all models considered on different models and dataset in table 2. Among these methods, *UserKNN* and *BMF* are one-step approaches that only predict overall ratings. *MultiUserKNN* is an extension of *UserKNN* which utilizes both overall ratings and multi-criteria ratings to compute similarities between user pairs. We can see that the accuracy of *MultiUserKNN* is slightly worse than the accuracy of *UserKNN*, which means *MultiUserKNN* is not an effective way to uncover the information of multi-criteria ratings.

However, not all multi-criteria based approaches perform better than the single-criterion based method *BMF*, though they exploit more information. For instance, the error of *MLR* and *SVR* is higher of *BMF*. These regression based methods initially predict multi-criteria ratings of the target item. However, this sub-process also imposes prediction error, which is amplified in the final stage and leads to poor predictions of overall ratings. Turning attention to the *MAE* metric, *CIC* achieves lower error than *BMF* on *Yahoo!Movies* dataset. However, *CIC* performs worse than *BMF* on remaining datasets. This implies that the performance of *CIC* do not generalize well on different applications. In addition, the deep learning base method (*DMCF*) achieve worse performance than our method (CFM_{ind}). This is because that the *DMCF* is not trained *end-to-end* [7, 25], which amplifies the error in its sub-process.

Among those methods, CFM_{ind} achieves the best performance, as it obtains the lowest error over all baselines. Although CFM_{user} and CFM_{item} take advantage of the transfer learning and share the latent vector of the user and item, they do not outperform the independent variant CFM_{ind}. When the target data is sparse, the knowledge sharing method may be helpful in improving the predictive accuracy of the target domain [21]. However, in the multi-criteria recommender system, the overall rating has the same sparsity with the multi-criteria rating. This means that sharing the user'(item') latent vector may dilute the knowledge in the domain of the overall rating, and therefore it is better to keep the latent space independent in the multi-criteria recommender systems.

Although CFM_{ind} outperforms *BMF* on three datasets, the improvements are various for different datasets. We can see that CFM_{ind} perform better than *BMF* on *BeerAdvocate* dataset. This is because users' overall ratings in *BeerAdvocate* have the low correlations with their multi-criteria ratings. According to equation (7), our method combines contributions of overall ratings and multi-criteria ratings in a linear manner, which leads to modest performance of our method on *BeerAdvocate* dataset.

Overall, our proposed CFMs obtain the best performance on all dataset, by achieving up to 10.52% and 13.14% lower RMSE and MAE than the state-of-the-art approach CIC.

Table 2: The performance of recommendation approaches. The standard error is presented in the bracket. Bold values indicate the best results.

	RMSE			MAE		
	TripAdvisor	*Yahoo!Movies*	*BeerAdvocate*	*TripAdvisor*	*Yahoo!Movies*	*BeerAdvocate*
UserKNN	1.2159(0.0239)	1.2329(0.0696)	0.8444(0.0150)	0.9458(0.0252)	0.9260(0.0716)	0.6559(0.0169)
MultiUserKNN	1.2146(0.0238)	1.2396(0.0715)	0.8441(0.0161)	0.9458(0.0255)	0.9319(0.0729)	0.6572(0.0177)
BMF	0.6820(0.0088)	0.8646(0.0061)	0.5858(0.0009)	0.4032(0.0023)	0.6289(0.0032)	0.4394(0.0001)
MSVD	0.9505(0.0596)	0.8738(0.0046)	0.5960(0.0006)	0.6387(0.0109)	0.6332(0.0030)	0.4473(0.0039)
MLR	0.7475(0.0081)	0.8664(0.0060)	0.5929(0.0002)	0.5255(0.0009)	0.6326(0.0066)	0.4442(0.0006)
SVR	0.7465(0.0086)	0.8671(0.0058)	0.5993(0.0021)	0.5109(0.0038)	0.6248(0.0063)	0.4470(0.0051)
CIC	0.6836(0.0140)	0.8782(0.0185)	0.5914(0.0070)	0.4055(0.0054)	0.6200(0.0055)	0.4429(0.0053)
DMCF	0.8289(0.0101)	0.9139(0.0078)	0.6240(0.0098)	0.5819(0.0028)	0.7012(0.0017)	0.4698(0.0058)
CFM_{user}	0.6492(0.0129)	0.8802(0.0095)	0.5904(0.0019)	0.3965(0.0013)	0.6184(0.0080)	0.4403(0.0001)
CFM_{item}	0.6549(0.0095)	0.8869(0.0035)	0.5904(0.0017)	0.3898(0.0021)	0.6145(0.0046)	0.4408(0.0008)
CFM_{ind}	**0.6117(0.0011)**	**0.8514(0.0063)**	**0.5833(0.0003)**	**0.3522(0.0065)**	**0.6042(0.0020)**	**0.4360(0.0004)**

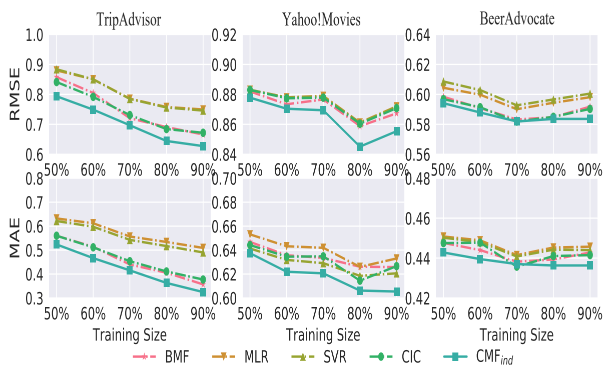

Figure 2: The performance of methods $w.r.t.$ **different training sizes.**

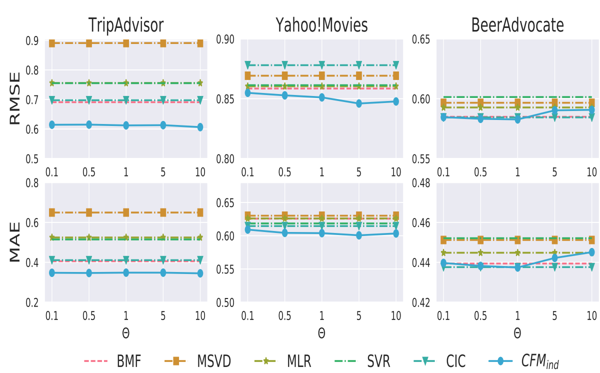

Figure 3: The performance of our method with different values of Θ.

5.3 The impact of sparsity and hyper-parameter

We study the performance of the model when the sparsity of the dataset varies, as shown in figure 2. The x-axis is the proportion of overall ratings in the training set to the total number of all overall ratings, where the y-axis is the error metric. In general, CFM_{ind} outperforms baseline approaches when the training size ranges from 50% to 90%, which shows the superiority of our method. On the *BeerAdvocate* dataset, CFM_{ind}'s *MAE* is slightly higher than *MAE* of *CIC* when the training size is 70%.

Our method adopts a linear method to combine the loss function of the overall rating and the multi-criteria rating (see equation (8)) and employs a hyper-parameter (Θ_α) to control contributions of multi-criteria ratings. We then evaluate the performance of our method with different values of Θ_α. For all criteria, we use the same value of Θ (i.e. $\Theta_1 = \Theta_2 = ... = \Theta_c$). The result is shown in figure 3. We can see that our method achieves the best predictive accuracy when Θ is around 1 for *TripAdvisor* and *BeerAdvocate* datasets. This indicates that individual multi-criteria rating has almost equal contribution with the corresponding overall rating. For the *Yahoo!Movies* dataset, the optimal Θ is around 5.

6 CONCLUSION

In this paper, we propose an end-to-end collective factor model (CFM) for the multi-criteria recommender system. Our methods integrate loss functions of overall ratings and multi-criteria ratings in a linear manner, such that both overall ratings and multi-criteria ratings are exploited to train the collective factor model. Our model does not need to estimate a user's multi-criteria ratings as a sub-process, which makes the system more robust than two-stages based methods. Experiment results on 3 benchmark datasets show that our methods outperform 8 different baselines, by achieving up to 10.52% and 13.14% lower RMSE and MAE than the state-of-the-art approach CIC.

ACKNOWLEDGMENTS

This research was supported in part by Key-Area Research and Development Program of Guangdong Province 2019B111103001, the China NSFC Grant (U2001207, 61872248, 61902249), Guangdong NSF 2017A030312008, Shenzhen Science and Technology Foundation (No. ZDSYS20190902092853047), the Project of DEGP (No. 2019KCXTD005), the Guangdong "Pearl River Talent Recruitment Program" under Grant 2019ZT08X603.

REFERENCES

[1] Gediminas Adomavicius and YoungOk Kwon. 2007. New Recommendation Techniques for Multicriteria Rating Systems. *IEEE Intelligent Systems* 22, 3 (May 2007), 48–55.

[2] Junhua Chen, Wei Zeng, Junming Shao, and Ge Fan. 2019. Preference modeling by exploiting latent components of ratings. *Knowledge and Information Systems* 60, 1 (2019), 495–521.

[3] Li Chen, Wei Zeng, and Quan Yuan. 2013. A unified framework for recommending items, groups and friends in social media environment via mutual resource fusion. *Expert Systems with Applications* 40, 8 (2013), 2889–2903.

[4] Yi Fang and Luo Si. 2011. Matrix Co-factorization for Recommendation with Rich Side Information and Implicit Feedback. In *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems* (Chicago, Illinois) *(HetRec '11)*. ACM, 65–69.

[5] Matthias Fuchs and Markus Zanker. 2012. Multi-criteria ratings for recommender systems: an empirical analysis in the tourism domain. In *International Conference on Electronic Commerce and Web Technologies*. Springer, 100–111.

[6] Guibing Guo, Jie Zhang, Zhu Sun, and Neil Yorke-Smith. 2015. LibRec: A Java Library for Recommender Systems. In *Proceedings of the 23rd Conference on User Modelling, Adaptation and Personalization (UMAP '15 Workshop, Vol. 4)*. 1–4.

[7] Haoran Huang, Qi Zhang, Xuanjing Huang, Haoran Huang, Qi Zhang, and Xuanjing Huang. 2017. Mention Recommendation for Twitter with End-to-end Memory Network. In *IJCAI*. 1872–1878.

[8] Dietmar Jannach, Zeynep Karakaya, and Fatih Gedikli. 2012. Accuracy Improvements for Multi-criteria Recommender Systems. In *Proceedings of the 13th ACM Conference on Electronic Commerce* (Valencia, Spain) *(EC '12)*. ACM, 674–689.

[9] Dietmar Jannach, Markus Zanker, and Matthias Fuchs. 2014. Leveraging multi-criteria customer feedback for satisfaction analysis and improved recommendations. *Information Technology & Tourism* 14, 2 (01 Jul 2014), 119–149.

[10] Naime Ranjbar Kermany and Sasan H. Alizadeh. 2017. A hybrid multi-criteria recommender system using ontology and neuro-fuzzy techniques. *Electronic Commerce Research and Applications* 21 (2017), 50 – 64.

[11] Pan Li and Alexander Tuzhilin. 2019. Latent Multi-criteria Ratings for Recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems* (Copenhagen, Denmark) *(RecSys '19)*. ACM, New York, NY, USA, 428–431.

[12] Qiudan Li, Chunheng Wang, and Guanggang Geng. 2008. Improving Personalized Services in Mobile Commerce by a Novel Multicriteria Rating Approach. In *Proceedings of the 17th International Conference on World Wide Web* (Beijing, China) *(WWW '08)*. ACM, 1235–1236.

[13] Jing Liu, Yu Jiang, Zechao Li, Zhi-Hua Zhou, Hanqing Lu, et al. 2015. Partially Shared Latent Factor Learning With Multiview Data. *IEEE Transactions on Neural Networks and Learning Systems* 26, 6 (June 2015), 1233–1246.

[14] Nathan N. Liu, Evan W. Xiang, Min Zhao, and Qiang Yang. 2010. Unifying Explicit and Implicit Feedback for Collaborative Filtering. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management* (Toronto, ON, Canada) *(CIKM '10)*. ACM, 1445–1448.

[15] Hao Ma, Haixuan Yang, Michael R. Lyu, and Irwin King. 2008. SoRec: Social Recommendation Using Probabilistic Matrix Factorization. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management* (Napa Valley, California, USA) *(CIKM '08)*. ACM, 931–940.

[16] Julian John McAuley and Jure Leskovec. 2013. From Amateurs to Connoisseurs: Modeling the Evolution of User Expertise Through Online Reviews. In *Proceedings of the 22nd International Conference on World Wide Web* (Rio de Janeiro, Brazil) *(WWW '13)*. ACM, 897–908.

[17] Angeliki Mikeli, Dimitris Sotiros, Dimitris Apostolou, and Dimitris Despotis. 2013. A multi-criteria recommender system incorporating intensity of preferences. In *4th International Conference on Information, Intelligence, Systems and Applications (IISA '13)*. IEEE, 1–6.

[18] Nour Nassar, Assef Jafar, and Yasser Rahhal. 2019. A novel deep multi-criteria collaborative filtering model for recommendation system. *Knowledge-Based Systems* (2019), 1–7.

[19] Mehrbakhsh Nilashi, Othman bin Ibrahim, and Norafida Ithnin. 2014. Hybrid recommendation approaches for multi-criteria collaborative filtering. *Expert Systems with Applications* 41, 8 (2014), 3879 – 3900.

[20] Mehrbakhsh Nilashi, Dietmar Jannach, Othman bin Ibrahim, and Norafida Ithnin. 2015. Clustering- and regression-based multi-criteria collaborative filtering with incremental updates. *Information Sciences* 293 (2015), 235 – 250.

[21] Weike Pan and Qiang Yang. 2013. Transfer learning in heterogeneous collaborative filtering domains. *Artificial Intelligence* 197 (2013), 39 – 55.

[22] Arkadiusz Paterek. 2007. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD cup and workshop (KDDCup '07, Vol. 2007)*. 5–8.

[23] Martin Saveski and Amin Mantrach. 2014. Item Cold-start Recommendations: Learning Local Collective Embeddings. In *Proceedings of the 8th ACM Conference on Recommender Systems* (Foster City, Silicon Valley, California, USA) *(RecSys '14)*. ACM, 89–96.

[24] Ajit P. Singh and Geoffrey J. Gordon. 2008. Relational Learning via Collective Matrix Factorization. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Las Vegas, Nevada, USA) *(KDD '08)*. ACM, 650–658.

[25] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end Memory Networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems* (Montreal, Canada) *(NIPS'15)*. Cambridge, MA, USA, 2440–2448.

[26] Hongning Wang, Yue Lu, and ChengXiang Zhai. 2011. Latent Aspect Rating Analysis Without Aspect Keyword Supervision. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Diego, California, USA) *(KDD '11)*. ACM, 618–626.

[27] Quan Yuan, Li Chen, and Shiwan Zhao. 2011. Factorization vs. Regularization: Fusing Heterogeneous Social Relationships in Top-n Recommendation. In *Proceedings of the Fifth ACM Conference on Recommender Systems* (Chicago, Illinois, USA) *(RecSys '11)*. ACM, 245–252.

[28] Wei Zeng, Meiling Fang, Junming Shao, and Mingsheng Shang. 2016. Uncovering the essential links in online commercial networks. *Scientific Reports* 6 (Sep 2016), 34292.

[29] Zhibin Zhang, Cong Zou, Ruifeng Ding, and Zhenzhong Chen. 2019. VCG: Exploiting Visual Contents and Geographical Influence for Point-of-Interest Recommendation. *Neurocomputing* (2019).

[30] Guoshuai Zhao, Xueming Qian, and Chen Kang. 2017. Service rating prediction by exploring social mobile users' geographic locations. *IEEE Transactions on Big Data* 3, 1 (March 2017), 67–78.

[31] Vincent W. Zheng, Yu Zheng, Xing Xie, and Qiang Yang. 2010. Collaborative Location and Activity Recommendations with GPS History Data. In *Proceedings of the 19th International Conference on World Wide Web* (Raleigh, North Carolina, USA) *(WWW '10)*. ACM, 1029–1038.

[32] Yong Zheng. 2017. Criteria Chains: A Novel Multi-Criteria Recommendation Approach. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces* (Limassol, Cyprus) *(IUI '17)*. ACM, 29–33.